



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

09/933,540

08/20/2001

Robert Barritz

P / 1318-136

5806

2352

7590

08/10/2005

OSTROLENK FABER GERB & SOFFEN
1180 AVENUE OF THE AMERICAS
NEW YORK, NY 100368403

EXAMINER

HA, THANH T

ART UNIT

PAPER NUMBER

2194

DATE MAILED: 08/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/933,540

Applicant(s)

BARRITZ, ROBERT

Examiner

Ha Thanh

Art Unit

2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 31 May 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-7 and 9-33 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-7, 9-33 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

22

DETAILED ACTION

1. Claims 1-7, 9-33 are pending in the application.
2. Claim 8 has been cancelled by applicant.

Specification

3. Applicant is reminded of the proper language and format for an abstract of the disclosure. The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details (i.e. Abstract too long).
4. Correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

5. Claims 1, 9, 11-12, 14-25 are rejected under 35 U.S.C. 102(e) as being anticipated by Lin et al. (hereafter Lin) (U.S. Patent 5949415).

6. As to claim 1, Lin teaches the invention as claimed including a system for determining program usage on a computer [abstract, line 2], the system comprising:

a plurality of executable software programs constituting software products, each of the software products being constituted of one or more load modules [abstract, lines 4-6], the load modules being stored in at least one memory of the computer [col. 4, lines 1-7];

an operating system of the computer that controls execution in the computer of the software products through the invocation of respective load modules thereof [col. 2, lines 39-45];

a monitor that is periodically triggered to collect load module execution information [col. 4, lines 10-11 and lines 19-21, col. 5, lines 38-41];

a filtering facility that is effective to filter known system programs from the load module execution information [col. 9, lines 49-53 & col. 10, lines 11-15].

a correlator that correlates ("match") the filtered load module execution information with data that associates load module names with corresponding software products [col. 6, lines 1-3] and develops ("builds") a list of software products executed in the computer over the course of a given time period [col. 6, lines 8-10];

a reporter that outputs data reflecting the use of the software products in the computer in terms of software product names thereof [col. 1, lines 66-67 and col. 6, lines 27-28].

7. As to claim 9, Lin teaches in the load module execution information module names and other process related information including directory, start time and process ID [col. 7, lines 46-68, and col. 8, lines 1-10].
8. As to claim 11, Lin teaches the correlator operates in conjunction with a knowledge base that associates load module names with software product names [col. 8, lines 56-57].
9. As to claim 12, Lin teaches a surveying program that develops an inventory of substantially all software products on the computer and a facility which produces a list of non-used software products based on comparing the inventory of software products against the data outputted by the reporter which reflects the use of the software products in the computer [col. 7, lines 26-33].
10. As to claim 14, Lin teaches the correlator operates by correlating module usage data with an inventory of software products that itself has been obtained by correlating a knowledge base load module names with software product names [col. 6, lines 1-3].

11. As to claim 15, Lin teaches the invention substantially as claimed including a system for determining program usage on a computer [abstract], the system comprising;

a plurality of executable software programs constituting software products, each software product being constituted in turn of one or more load modules, the load modules being stored in at least one memory of the computer [col. 4, lines 1-3];

an operating system of the computer that controls execution in the computer of the software products through the invocation of respective load modules thereof [col. 2, lines 37-45];

a monitor that collects load module execution information by deducing which load modules are being used in given processes of the computer, without directly monitoring the actual invocation by the operating system of the load modules [col. 2, lines 46-48];

a filtering facility that is effective to filter known system programs from the load module execution information [col. 9, lines 49-53 & col. 10, lines 11-15].

a correlator that correlates ("match") the filtered load module execution information with data that associates load module names with corresponding software products [col. 6, lines 1-3] and develops ("builds") a list of products executed in the computer over the course of a given time period [col. 6, lines 8-10]; and

a reporter that outputs data reflecting the use of the software products in the computer in terms of software product names thereof [col. 6, lines 27-28; and col. 1, lines 66-67].

12. As to claim 16, Lin teaches the monitor obtains the load module execution information from a load module table ("local record file" col. 4, lines 10-11] created by the operating system which makes entries in the load module tables as processes are executed and access requests for load modules are made [col. 4, lines 19-21].

13. As to claim 17, Lin teaches the monitor is implemented to execute every time the end of a process is reached [col. 4, lines 36-41].

14. As to claim 18, the monitor executes as an exit routine near the end of a process [col. 4, lines 36-41].

15. As to claim 19, the monitor gathers and accumulates usage data across sub processes of a higher level process so that when the load module table is successively read, only those module entries not previously encountered in a prior sub process of the current high level process are accumulated and names of load modules already found the table for the current high level process are ignored [col. 5, lines 34-41].

16. As to claim 20, the correlator operates by identifying the names of all software products used by correlating module usage data by using a knowledge base that associates the names of load modules with software products they comprise [col. 8, lines 56-57].

17. As to claim 21, the correlator operates by correlating module usage data with an inventory of software products that itself has been obtained by correlating in a knowledge base load module names with software product names [col. 6, lines 1-3].

18. As to claim 22, A system for determining program usage on a computer [abstract], the system comprising:

a plurality of executable software programs constituting software products, each of the software products being constituted of one or more load modules, the load modules being stored at least one memory the computer [col. 4, lines 1-3];

an operating system of the computer that controls execution in the computer of software products through the invocation of respective load modules thereof [col. 2, lines 37-45];

a monitor that collects software product execution information by monitoring input or output to specific files or datasets by the software products [col. 5, lines 49-55], and

a filtering facility that is effective to filter known system programs from the load module execution information [col. 9, lines 49-53 & col. 10, lines 11-15], wherein such inputs and outputs are associated with corresponding software products or groups of software products [col. 5, lines 49-55].

19. As to claim 23, Lin teaches the monitor is implemented as a background process [col. 4, lines 37-40].

20. As to claim 24, Lin teaches the monitor is implemented as an intercept systematically placed either or both of a file open and/or file close system function of the computer [col. 4, lines 37-41].

21. As to claim 25, Lin teaches the monitor is operated as a batch process [col. 3, lines 58-60].

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

22. Claims 2-7, 10, 13, 26-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lin et al (hereafter Lin) (U.S. Patent 5949415), in view of Johnson (U.S. Patent 6788980).
23. As to claim 2, Lin teaches the monitor perform certain processing on a periodic basis such as taking the current system time [col. 6, lines 35-38]. Lin does not specifically teach the monitor operates by taking periodic snapshots of the current state of active processes in the computer. However, Johnson teaches taking snapshot to be capable of retrieving the values all at once [col. 22, line 28]. It would have been obvious to one of ordinary to combine the teaching of Johnson's and Lin's to have the monitor takes snapshots of all active process because Johnson's teaching would enhanced the monitor's performance by keeping track of more system information than merely the current system time.
24. As to claim 3, Lin teaches the load module execution information includes one or more of the following data items: module names, user names, the time when processes were started, how much CPU time has been used, and a directory path name from which processes were installed [col. 7, lines 46-68 and col. 8, lines 1-10].
25. As to claim 4, Johnson teaches a facility that allows adjusting the period between snapshots in response to program usage activity level [col. 22, lines 20-23].

26. As to claim 5, Lin teaches the monitor produces a list of executing load modules and their respective directory path name [col. 8, lines 4-6].

27. As to claim 6, Johnson teaches a facility that determines how many processes have begun and ended between snapshots [col. 22, lines 27-28].

28. As to claim 7, Johnson teaches a facility that compares successive snapshots to determine which modules have executed and how many were missed [col. 22, line 30-31].

29. As to claim 10, Lin teaches processes are identified by PID (Process Ids) ["LaunchID" col. 5, line 6] and the monitor includes a facility that obtains a measure of missed processes [col. 10, lines 34-37].

30. As to claim 13, Lin does not specifically teach the knowledge base is a database of records which also associates file names to software products that use them and additionally includes at least one of the following: flags indicating if a module is used uniquely or shared among vendor products; a number indicating file matches required for correlation with the product; file type; file size; file creation date; and embedded strings of text.

31. However, Lin teaches a "database with program usage information, identifying in a manner which uniquely associates it with client computer" and "to separately maintain program usage information for a plurality of clients" [col. 3, lines 17-22].

32. It would have been obvious to one of ordinary skill in the art to apply Lin's teaching to keep database of records of file names to software products that used them and any additional information that is useful for the maintenance of the product.

33. One would have been motivated to apply the method of Lin for associating program usage with specific clients to associate file usage with specific software products to advantageously capture those relationships for file and system maintenance and updating/upgrading as well as to monitor and determine the usage of key files, used by specific software products.

34. As to claim 26, Lin teaches a system for determining non-usage of software products on a computer, the system comprising:

a plurality of executable software programs constituting software products, each of the software products being constituted one or more load modules, the load modules being stored in at least one memory of the computer [col. 4, lines 1-3];

an operating system of the computer that controls execution in the computer of software products through the invocation respective load modules thereof [col. 2, lines 39-45];

a software product surveyor that surveys the at least one memory of the computer and produces an inventory of the names the load modules, the surveyor being operable with an associator that identifies and associates and records associations between product names and the load module inventory names [col. 6, lines 1-3, and lines 8-10];

a monitor that collects load module execution information over a given time period [col. 4, lines 10-11, and lines 19-21;and col. 5, lines 38-41];

a filtering facility that is effective to filter known system programs from the load module execution information [col. 9, lines 49-53 & col. 10, lines 11-15];

a correlator that correlates ("match") the filtered load module execution information with data that associates load module names with corresponding software products [col. 6, lines 1-3] and develops ("builds") a list of products executed in the computer over the course of a selected time period [col. 6, lines 8-10];

Lin does not specifically teach a comparing facility that compares information provided by the correlator to information provided directly or indirectly by the surveyor and which produces a list of unused software products; and a reporter that outputs data reflecting the non-use of software products in the computer. However, Lin disclosed program monitor that pools operating system looking for

active task [col. 7, lines 25-27] and obtains start and stop information by deducing this information from the present and absence of active tasks [col. 7, lines 30-33]. It would have been obvious to one of ordinary skill in the art to recognize that by having usage information, one can deduces to have non-usage information. One would have been motivated to keep track of non-usage information of software products to delete non-used products from the system thereby cutting cost on software maintenance and also saving system space.

35. As to claim 27, Lin teaches the comparator is constructed to delete from the inventory those software product names which have been detected by the monitor as having been executed in the computer at least once during the given time period, leaving a list non-used software products [col. 7, lines 17-19].

36. Claims 28-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lin et al (hereafter Lin) (U.S. Patent 5949415), in view of Evans (U.S. Patent 6430708).

37. As to claim 28, Lin teaches the invention substantially as claimed including a system for determining program usage on a computer [abstract], the system comprising:

a plurality of executable software programs constituting software products, each of the software products being constituted of one or more load modules, the

load modules being stored in at least one memory the computer [col. 4, lines 1-3];

an operating system of the computer that controls execution in the computer of software products through the invocation of respective load modules thereof [col. 2, lines 37-45];

a monitor that collects load module execution information reflecting the usage of software products on the computer [col. 5, lines 38-40];

a filtering facility that is effective to filter known system programs from the load module execution information [col. 9, lines 49-53 & col. 10, lines 11-15];

a reporter that outputs data showing respective directory paths for load modules that have been executed [col. 8, lines 4-6];

Lin does not specifically teach a library source determination facility that determines the load library from which each executed load module has been loaded. However, Evans teaches "load libraries that hold load modules needed to execute the batch programs" and said "these types of libraries are well known in the art" [col. 7, lines 12-15]. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined Lin and Evans's teaching of using the library because by organizing the executable module in a libraries it would provided more accessibility, therefore improve performance of the system.

38. As to claim 29, Lin teaches the library source determination facility obtains a list of modules that have been used by a particular process, determines the load libraries and their search order used by the process, and using a search order determined in a prior step, searches the load libraries of the computer for a first library containing the same modules that best matches the list of modules used [col. 6, lines 17-19].

39. As to claim 30, Lin teaches the monitor operates as a concurrent process and module usage data and load library collection data are both obtained by the monitor and library usage is concurrently determined [col. 10, lines 57-60].

40. As to claim 31, Lin teaches the task of determining the correct load libraries their appropriate search order is carried out as a separate process, wherein one of the module usage data or library collection data is obtained by the monitor and the other is obtained from a separate source and processed to determine load library usage ["separately maintain program usage information" col. 3, line 21].

41. As to claim 32, Lin as modified teaches the library source determination facility uses a JCL (Job Control Language) interpreter [Evans, col. 5, lines 40-42].

42. As to claim 33, Lin as modified teaches the load library determination facility determines both the identity and order of load libraries used by a particular

process by reading job control language ("JCL") data structures current job to obtain a load library list for the process [Evans, col. 6, lines 45-60].

Response to Arguments

1. Applicant's arguments filed 05/31/2005 have been fully considered but they are not persuasive.
2. In the remarks, Applicant argued in substance that Lin does not teach or suggest the
 - a. "filtering facility" and
 - b. the "correlator that correlates the filtered load module execution information with data that associates load module names with corresponding software products".
3. As to point (a)- Lin clearly discloses a filtering facility ("detecting step", "extracting an identifier") that is effective to filter known system programs from the load module execution information [col. 9, lines 49-53 & col. 10, lines 11-15] ("the operating system maintains a system task list identifying active tasks and subtasks, said detecting steps comprising interrogating the system task list and extracting an identifier for an active task or subtask").

4. As to point (b)- Lin clearly discloses a correlator that correlates ("match") the filtered load module execution information with data that associates load module names with corresponding software products [col. 6, lines 1-3] ("by searching the application records in local record file for a record having a TaskID matching the task identifier in the callback message") and develops ("builds") a list of software products executed in the computer over the course of a given time period [col. 6, lines 8-10] ("builds a new task record in local record file using the appropriate TaskID and LaunchID");
5. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ha Thanh whose telephone number is 571-272-7220. The examiner can normally be reached on 8:00 AM - 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).


MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2194